JORDAN ZLATEV

Criteria for Computational Models of Morphology: The Two-Level Model as an NLP Framework

Abstract

Computational models of morphology are best seen not as morphological models but rather as natural language processing frameworks which can express descriptions in the style of one morphological model or the other, and even go further, but without necessarily being bound by "purely" theoretical considerations. Criteria for their adequacy can be derived by treating them (together with the linguistic descriptions that are expressed in their formalisms) as NLP systems, for which a number of goals can be stated, among which are sufficient coverage, efficiency, augmentability and flexibility. The two-level model (TWOL) of Kimmo Koskenniemi is the main object of attention in this article and examples of its applicability to Bulgarian morphology are presented.

1 Introduction

The criteria for what a "morphological model" should be able to account for, and the manner in which this should be done, have risen high during the past few years in accordance with the situation in the neighbouring linguistic "levels" of syntax and semantics. Apart from the traditional requirements for *linguistic felicity* ("capturing the generalizations"), *rigour*, and *simplicity*, opinions are being expressed that a morphological model should be *general*, (understood as universal), *explanatory* and even *psychologically real*. Now far from doubting the plausibility of these requirements, I feel that they tend to place the models of human language provided by the field of computational linguistics in a rather unfavourable light. This is especially relevant for computational morphology, which only during this decade seems to have "stepped out from the cradle", as for example Lars Borin (p.c.) has implied. And instead of being blindly critical and sceptical towards its potentials, (which "linguists proper" often tend to be

Proceedings of NODALIDA 1989, pages 86-100

towards computational linguistics in general, as a form of self-defence), isn't it best to watch its first steps carefully, with a helpful hand where it can be lent?

If in the previous paragraph I have suggested the picture of computational morphology, and more concretely of it's best known representative, the *two-level model*, (first presented in (Koskenniemi 1983) and most often abbreviated *TWOL*), as a clumsy, stumbling baby, then I have gone too far in my manner of expression. Nothing can be further from the truth considering the enormous amount of attention and subsequent work that Koskenniemi's dissertation unleashed. Hardly a conference can go by—including this one—without a few contributions pointing out TWOL's achievements—or deficiencies, and in the best case offering improvements or alternatives, e.g. (Sproat and Brunson 1987, Bear 1988, Kataja and Koskenniemi 1988, Calder 1989). But even these can without doubt fall at the hand of the theoretical linguist who will not fail to see the inadequacy of Bear's reintroducing the notion of "negative rule features", or of Calder's "string equations". As the last author himself carefully states: "... one may justifiably have reservations about introducing string equations into linguistic descriptions." (Calder 1989:62).

In this paper I wish to propose what I think is a more "constructive" view of the aims of computational morphology, which is also more or less applicable to the field of (computational) natural language processing in general. I will argue that there are a number of properties, which can help us compare, evaluate and develop models in a more short-term perspective so that one need not necessarily be overwhelmed by the "theoretical argument" from the beginning. In my opinion a computational model that finds the best combination of these properties, has also the best chances of being theoretically significant as well, though this is a somewhat controversial matter. A viewpoint that is at least **less** controvercial is that the goals of computational and theoretical linguistics differ. Shieber (1987), for example, has claimed that these differences, especially concerning "restrictiveness", are so essential that from a computational perspective one is more interested in **what** the linguistic theories say than **how** they say it and that it is meaningful to try to separate theories ("how") from their analyses ("what") and concentrate on the latter in computational models.

I would like to continue on this line of thought with one substantial difference: while Shieber discusses models in their property of being "computer tools for linguistics", I regard them as potential candidates for becoming language theories on their own. This difference is illustrated in the choice of model to exemplify the issues under discussion: in Shieber's case this is the formalism of PATR-II, while I will use the two-level model. I will be presupposing at least **some** previous knowledge of it.

2 Computational Models as NLP Frameworks

I believe that one could say that the aims of computational and theoretical linguistics eventually converge, namely to gain a better understanding of the nature of human language and of its user. Still they differ in their methods.

Computational linguistics (partly because of utilitarian reasons) is much more inclined to use the trial-and-error approach, starting with a fragment and then augmenting it; taking some categories for granted, (phonemes, for example) as "working hypotheses", if they facilitate the overall work of the system. This is so because the short-term goal of computational linguistics is the construction of a *natural language processing system*, no matter if it does or does not model human language processing at a sufficiently theoretical level. On the other hand it is theoretical linguistics that should stand for the "conceptual insights", the new ideas and the quest for linguistic universals. Of course, the closer the connection between computational and theoretical linguistics, the better, but at least to begin with, this is **not** a necessity.

What I'm aiming at is to say that computational and theoretical models should not be considered *on a par*. A computational model is both less and more than a theoretical one. Less, because it is the backbone of a system and thus is subjected to the limitations I mentioned above, i.e. working hypotheses, fragments etc. More, because if it is flexible enough it could permit **several** theoretical models to be implemented (simulated) within it. So the question whether TWOL is a morphological model or not, is not all that relevant. As to whether it is "general" and in what sense, I will come to that later. Right now an important (in my opinion) question arises, namely:

If at least the short term aims of computational and theoretical linguistics split, then what are to be the criteria for, let us say "evaluating", computational models (theories, formalisms—the terminology varies) for morphology in particular, and natural language in general? There is no simple answer to this question. As a half year's survey of the relevant literature, reported in (Zlatev et al. 1989) and (Sågvall-Hein et al. 1989), has managed to convince us—opinions differ. We came to believe that in order to come to more abstract things such as desiderata, requirements etc. for the models, one should start with something more concrete. The key lies in what I mentioned above was one of the first aims of computational linguistics, and definitely the first of its more practically oriented sub-branch, Natural Language Processing (NLP): the creation of an *NLP system.*

Now what kind of animal is that? This need hardly be defined for "insiders", but for someone unfamiliar with the jargon in the field, it should be enough to say that an NLP system can be regarded as a unity of (at least) the following elements: (1) an implementable formalism, (2) a processing mechanism, and (3) linguistic knowledge expressed in the formalism. (1) and (2) together make up the computational model or—using a term more neutral to the computational/theoretical dichotomy which I myself introduced—an *NLP framework.* (3) is the *language description.* The three are as I said interdependent, but to different degrees in different systems.

3 Viewing TWOL as an NLP Framework

The main advantage in viewing computational models of natural language and of morphology in particular as NLP frameworks comes from the fact that it is possible to formulate relatively clearly what goals NLP systems should aim at. Then one could continue "bottom-up" to state "criteria" on how the models should be shaped in order to correspond to these goals. Consequently these are criteria of a practical nature which are not "theoretically bound" to begin with. Most interestingly, however, they have implications which are highly compatible with linguistically motivated considerations. I will come to this in the last section.

A computational model such as TWOL may be seen as providing the framework for an NLP system. It still remains to be "filled" with the concrete linguistic knowledge. Now the first question that arises is: how much knowledge can be expressed in the framework? The first goal for an NLP system is that this knowledge is sufficient for current purposes, or alternatively formulated, that it has sufficient coverage.

3.1 Sufficient Coverage

If a description of a certain fragment of one or several languages can be made so that the system "works" as intended with respect to this fragment, then the framework can be regarded as expressive enough in relation to this fragment. Thus one may say that an NLP framework is *weakly complete* (in Shieber's terminology) if and only if it provides a system with the linguistic coverage necessary for the given purposes.

TWOL has been applied to substantial fragments of the inflectional morphology of a number of languages ranging from Finnish (1983) to Japanese (Alam 1983) and Old Church Slavonic (Lindstedt 1986). Now while this implies that the TWOL-framework is *general* in the sense that it has a **potentially** large coverage, it does not mean that TWOL is "general" in the sense that it can be applied to all of the world's languages and their morphology - inflectional and derivational (where this distinction exists), i.e. that it is a universal morphological model. It is rather a matter of degree: TWOL is "better" than most other models because it has been applied to larger fragments of single languages, e.g. "an (almost) full description (of all the forms of all inflectional types)" (Koskenniemi 1983:125) and because it has been applied to more languages. But then, what more is needed? The fact that the morphology of for example Kubachi (cf. Johannessen, this volume) yields difficulties, doesn't make TWOL a less suitable framework for the description of, let's say, Bulgarian inflection. This only means that the morphologies of the two languages are different—the opposite would be surprising. However, if one by a "general" framework means one that can provide adequate descriptions for all language types: agglutinating, isolating, inflecting, etc. then more is to be desired. This falls, in my opinion, not under the goal of coverage but of flexibility, which will be discussed further on.

Let us be more concrete. In (Zlatev 1988) I have given what I think is a complete description of Bulgarian nominal inflection in terms of the original

TWOL, i.e. as presented in (Koskenniemi 1983). Bulgarian morphology is very well developed and poses some non-trivial problems for any linguistic description, computational or not, such as extensive allomorphy and morphophonemic alternations within the stems. TWOL has proved quite satisfactory in describing both, with its finite-state lexicon and two-level rules, respectively. The demonstrative pronouns, however, dispay an "irregular" internal inflection, which in the original (Pascal) format of the lexicon gives no other opportunity for description than the following, which is far from elegant,

```
t          o-a-ov-e/P      "PRON DEM IDENT"
```

with 't' as the "invariant stem" (I have sticked to the principle: "One entry per Stem" so as to avoid masking some problematical areas through listing) and the continuation class `o-a-ov-e/P` which is the name of a mini-lexicon with the following content:

```
LEXICON o-a-ov-e/P  ozi  #    "MASC SING";
                    azi  #    "FEM SING";
                    ova  #    "NEUTR SING";
                    ezi  #    "PLUR"
```

If this had been the regular pattern for inflection in Bulgarian, then a possible computational description in the form of a system of intersecting lexicons—as those presented in (Kataja and Koskenniemi 1988) for the non-concatenative morphology of Semitic languages—would have been necessary (and probably sufficient). However, since the number of mini-lexicons of the kind shown above is *5* altogether and all other types fall neatly into the finite-state pattern, a compromise seems to be the best solution: I consider TWOL expressive enough, i.e. sufficient for current purposes, and decide to leave the description at that.

What if I decide to treat derivational morphology as well? Five classes of Bulgarian pro-forms seem to be readily describable as a derivational pattern which is something of the sort:

```
                          | INTERROGATIVE | + to = RELATIVE
                          |
INDEFINITE    = nÄ    + |   kakyv
NEGATIVE      = ni    + |   koga
GENERALIZING  = vsÄ   + |   kak
                          |     :
                   A          B               C
```

That is, the interrogative pro-forms (B), act as the "base", which together with the appropriate "prefix", build respectively indefinite, negative and generalizing pro-forms (A), and with the "suffix" 'to' (which is actually the postponed definite article for nouns and adjectives of neuter gender)—relative pro-forms (C). However, if we try to express this simple pattern in a finite-state lexicon then we will also derive ungrammatical word-forms such as **nÄkakto*, **nikakyvto* etc., i.e. overgeneration. The reason is that if a finite-state mechanism allows

AB and BC, then it must also allow ABC, which in this case we want to forbid. Similar problems with the English prefix un-, are discussed in (Karttunen and Wittenburg 1983).

Now does this mean that we have found a point where TWOL is not sufficient in terms of coverage and an argument that it is inapplicable to Bulgarian as well as possibly the derivational morphology of most languages?

To some extent—yes. For practical purposes we may double the entries of type B in the lexicon, so that we have B' and then connect the mini-lexicons, to get AB and B'C (for example). But this is a kind of "solution" that would lead us back to where we started, and it is in some sense even worse than listing the different word-forms—it is absurd that we should have to go all this way only to start duplicating entries. (Here I'm not concerned with matters of efficiency—but these are of course more than relevant as well.)

There are, however, two other much better ways out. One would be to replace the finite-state lexicon component with a phrase-structure one, which furthermore can use a feature-matching (unification) mechanism which would guarantee that only the grammatical forms are generated. For example the problem I mentioned above can be resolved the following way:

```
(1)  PRO(IND)   -->  nÄ  + PRO(INT)
     PRO(NEG)   -->  ni  + PRO(INT)
     PRO(GEN)   -->  vsÄ + PRO(INT)
     PRO(REL)   -->  PRO(INT) + to
```

An alternative—without increasing the expressive power of the formalism—is to use the model's two-level rules in order to block out ungrammaticalities. In the case above one must use at least two "diacritic characters", let us say, `@` and `#` (which must be clearly defined as bearers of morphological features and have nothing to do with phonology) and associate them with the entries of type A and C, respectively. Then a rule can be stated which would prevent their co-occurrence, (the operator `/<=` means "is disallowed" and what follows is the context which characterises all Bulgarian interrogatives, followed by the "relative sign"):

```
(2)  @   /<=  _ k V C (V) (C) #
```

Both (1) and (2) should have the same effect, and which one would be preferred is largely a matter of how they influence the goals to be discussed below, namely efficiency and augmentability.

3.2 Efficiency

Efficiency is something that concerns not only NLP systems for practical purposes, but theoretical ones as well, since all interesting applications of computational techniques to natural languages involve fragments that go beyond vocabularies of several hundred words and a predetermined number of sentences.

It is not hard to believe that it is just this criterion that has been the main reason for TWOL's popularity rather than its linguistic characteristics. The *restrictiveness* of the formalism gives the opportunity of extremely efficient implementations in which the lexicon has the form of a word tree and the rules—finite-state transducers. This has brought about the possibility of constructing systems with lexicons of tens of thousands of stems which can process text corpora and return analyses with morphological features at a spead of up to 100 word-forms a minute (Fred Karlsson p.c.)

I don't intend to indulge in this matter since I'm no expert. Still, the interdependency of system efficiency and other goals must be pointed out. For example, when it comes to choosing between the two solutions to the coverage problem which I discussed above, one would probably adopt the second alternative—that with the disallowing rule—if abandoning the finite-state format of the lexicon is likely to slow down implementations drastically. And this would be a reasonable move—as long as it doesn't get in the way of the next goal.

3.3 Augmentability

An NLP system is said to be augmentable if it can be improved with regard to each of its subcomponents—formalism, processing mechanism and linguistic description. Even if the first two are far from unchangeable, they are nevertheless more stable than the third, the development of which is by its nature an incremental and interactive process, which goes through loops, dead ends, partial solutions, gradual generalizations etc., until it reaches a provisionally stable level, and then again must be such that it is possible to develop it when the need arises. For this reason it is central that the formalism is *perspicuous*—a quite informal criterion, but nevertheless an important one.

I have already discussed questions pertaining to the TWOL lexicon component so I will take a few examples from the "heart" of the model—the two-level rules. Before a compiler for them existed, it was a cumbersome affair to translate into transducers even the simplest rules. The TWOL compiler (Karttunen et al. 1987) was a great advance in this respect. It gives the opportunity for a quite general morphophonemic rule to be formulated as simply as this, (`FrontV` is defined as `I,E.`):

```
(3)  "Palatalization of Velars"
     Cx:Cy <=> _(V:0) FrontV:
               where Cx in (k g x)
                     Cy in (c z s)
               matched ;
```

The reading is (for one unfamiliar with the notation): "lexical k is realized as surface c; lexical g as surface z and lexical x as surface s, if and only if they are followed by an optional lexical vowel which is realized as nothing (because of another rule), and a lexical front vowel (realized as anything on the surface level)". It will account for example for the following pairs. (The second example shows how productive the rule is!)

Lexical representation: `r y k a E   (hand + PL)`
Surface representation: `r y c   e`

Lexical representation: `h o t d o g  I   (hotdog + PL)`
Surface representation: `h o t d o z  i`

One of the reasons why such rules are more perspicuous than for example rules of generative phonology is that they are purely *declarative* statements which need not take into consideration any requirements of ordering and the complex interactions that go with it. It is first after compilation (into finite-state transducers) that they gain their procedural interpretation.

Another factor that makes it easier for a TWOL system to be augmented is the lexicon/rules distinction itself, which is an example of the positive consequences of *modularity.* For example even if new inflectional and derivational types are eventually discovered, the lexicon can be changed, but if a rule—as the one above—is general enough, then it need not be "tinkered with" at all, but can safely apply on the new lexical information.

Now, the rule as I stated it above is actually different from that in (Zlatev 1988) in that it does not use any diacritic characters which have the function of "morpheme boundaries", "triggers", "blockers" etc. The point is that such characters work against the perspicuity of the formalism and thus against the augmentability of the system. For example in (Zlatev 1988:33) I wrote: "The operational lexicon can be augmented with new lexical stems which only have to be given the appropriate continuation classes (and if necessary to use the diacritics in the right positions)". It is just these "right positions", which would make it so hard for anyone else than myself to develop the system.

3.4 Flexibility

The goal of flexibility is close to that of augmentability discussed above, but concerns the ability of change not only for the sake of improvement, but as a value in itself.

There are different levels of flexibility. One is the hardware dimension: a system should preferably be independent of any particular type of machine. The recent "emancipation" of the TWOL compiler from the demanding environment of Lisp-machines, i.e. the existence of a compiler running on the more powerful models of Apple Macintosh (Kimmo Koskenniemi p.c.) may be considered in this respect as a step in the right direction.

Another aspect of flexibility is with regard to software: a system should not in any major degree depend on a particular programming language. The fact that modern programming languages have equivalent absolute expressive power, i.e. that they are Turing equivalent, doesn't mean that they are functionally and notationally so (cf. Shieber 1987) and it is sometimes easy to fall for the "procedural seduction" of computational linguistics that Kaplan (1987) discusses, e.g. to depend on PROLOG's backtracking mechanism or on Lisp's evaluation procedures. TWOL has been implemented in Pascal (Koskenniemi 1983), Common Lisp (Gajek et al 1983), Interlisp-D (Dalrymple et al 1987) and C (Kimmo

Koskenniemi p.c.) which is a sign that the model is more or less independent of programming environment. Both these aspects of flexibility call for treating *hard- and software independence* as a criterion in itself.

Matters of implementation, however, are not of primary interest for us here. I would actually want to "stretch" the concept of flexibility of an NLP system and interpret is as *linguistic flexibility*, or in other words: the property of allowing different styles of description. It is here that the relevance of theoretical considerations is greatest. Let us look again at TWOL in a little more detail.

As familiar, TWOL can provide morphological descriptions in the style of both of the traditional morphological models "Item and Arrangement" (IA) and "Item and Process" (IP). For example considering rule (3) and the examples in 3.3. one can describe the singular and plural form of the Bulgarian lexeme 'hotdog' in the following way (assuming the feature-value format of the lexicon presented as an option in (Dalrymple et al 1987), though without the facilitating device of "templates"):

```
(4)     hotdo [[semantics: [meaning: 'hotdog']]
               [syntax: [cat: n]
                         [continuation: G/Z]]].
       LEXICON G/Z
           g  [[semantics: [num: sg]]
               [syntax: [continuation: #]]].

           z   [syntax :[continuation: /i]].

       LEXICON /i
           i  [[semantics: [num: pl]]
               [syntax: [continuation: #]].
```

and alternatively:

```
(5)  hotdog [[semantics: [meaning: 'hotdog']]
                [syntax: [cat: n]
                          [continuation: /I]]].

       LEXICON /I
           I  [[semantics: [num: pl]]
               [syntax: [continuation: #]].
```

While (4) makes use only of the lexicon, (5) relies on the "Palatalization of Velars"-rule as well (plus a default mechanism stating that *num* recieves the value *sg*, if unspecified).

If we have to compare the two alternatives, (5) seems to be better in almost all respects. It is not only more "elegant", it is shorter, simpler and as I argued in the previous section this type of description is more perspicuous and modular, thus a system based on it would be more easily augmented. From a linguistic perspective (4) would simply describe 'hotdog/hotdoz' as allomorphs in complementary distribution where only the second takes plural which is obligatory while (5) would furthermore incorporate the process of palatalization in the description and thus "explain" the allomorphy.

This could possibly imply that Bulgarian morphology (and probably any morphology with morphophonemic alternations) is more readily describable in terms of IP than IA. This is equivalent to saying that it could be associated with a "typological parameter" which determines the most appropriate style of description (cf. Matthews 1974:163).

However, the fact that this is an option in TWOL, which also permits descriptions of type (4)—supposedly sufficient for purely agglutinating languages—is an obvious advantage in terms of flexibility.

So TWOL can in practice fully "model" both IA and IP. The extent to which this is so is sometimes overlooked because of the fact that all implementations that I know of have used input such as `book + s`, instead of `book + PL` during generation, i.e. neglelected the information in the lexicon. This should not be considered a disadvantage of the model itself, since the only reason why it hasn't been implemented is that up to now TWOL has not been used for word-form production in any larger application. Morphological conditions are furthermore expressible (and are expressed all the time) in the contexts of the rules through the diacritic signs and "morphophonemes"—just as in IP.

What about "Word and Paradigm" (WP)? The fact is that if TWOL would also be flexible enough to permit descriptions of type WP, this would improve the model in terms of perspicuity to a considerable extent. I mentioned in the previous section that for the sake of the latter, diacritic characters are best avoided. However, in eliminating the "morpheme boundary" (e.g. `+` or `-`), I had to specify that the front vowels the suffixes start with do not belong to the stem and I did that in usual manner—by using uppercase letters (i.e. `I` and `E`), which under the popular terminology usually go as "morphophonemes". Now as for example Nyman (1988) points out, these are not internal to the model in any theoretical aspect, but simply express a convenient way of encoding morphological information in segments (e.g. `I = i + PL` etc.) and are necessary because the two-level rules, or rather the finite-state transducers they are compiled in, operate only on segments.

Now looking at (5) above, we can furthermore see that this information is redundant, since `I` is specified as a plural suffix in the lexicon as well. Furthermore there must be an "ordinary i" plural suffix for adjectives such as "plax" (frightful), which do not undergo palatalization in plural form (plax/plaxi).

A possible way to preserve the efficiency that processing segmental representation provides, while avoiding inconsistencies and improving clarity would be to incorporate a WP element in the TWOL formalism. This could possibly look the following way:

```
(6)  hotdog {[[semantics: [meaning: 'hotdog']]
                [syntax: [cat: n]
                         [continuation: /i]]]
              (paradigm: 'HOTDOG')}.

     LEXICON /i
          i  [[semantics: [num: pl]]
               [syntax: [continuation: #]].
```

The idea is that 'HOTDOG' can be defined (for example under the heading PARADIGMS) as a prototypical entry for a paradigm which undergoes an alternation—it need not be specified which, the information for this is contained in the rule—in **plural form**. Now the simplest way to implement this would be something like:

```
(7)  TRANSFORM(entry (continuation))
               IF (num (continuation)) = pl
```

On the other hand the set which the palatalization rule referred to could be redefined, so that it says instead:

```
(8)   FrontV = i + pl, e + pl.
```

Now if these can be compiled together, then one should be able to (though I haven't figured out a general mechanism yet) associate the value of TRANSFORM with the value of respectively `i + pl` and `e + pl`. A trivial way to do this would be to express the first after compilation as `I`, the second as `E` and TRANSFORM would then only have to be a function such as UPPERCASE. In this way the result would be the same as with "morphophonemes" but on a level that more clearly belongs to the "machinese" (cf. Nyman 1988) than the present formalism. The major gain from such a strategy would be that one would not have to worry about using uppercase letters "in the right positions", or to assign different continuation classes for the same suffix—as long as one gives the right "paradigm", which seems intuitively a much easier thing to do and, at the same time, is more linguistically motivated.

If this, or some similar mechanism, existed at least as an option in TWOL (which as yet it does not), then most probably the flexibility of the overall system would be increased in a way which would also permit faster development. This could be described as a matter of economy, which may also be considered as a goal in itself, albeit mainly for practical systems.

3.5 Economy

If by "economy" we mean that both the construction and the running of a system should not be too expensive in terms of time and work then one could say that all the criteria I have discussed in the previous sub-sections, i.e. sufficient generality, perspicuity, modularity etc. can be regarded as relevant. One may further point out that for the sake of economy is is much more practical to have a system that analyses and generates (if both are required, of course) with the same program, than to use two different ones, and this can only be achieved if the NLP framework is *bidirectional.*

In this respect, too, TWOL "scores a point", because the finite-state transducers are actually **correspondences**, not transformations, and it is as easy to go from lexical to surface form, as in the opposite direction. One drawback in all implementations up to now, as I mentioned above, is that the lexicon has only been equipped for recognition, but this should be amendable.

3.6 Psychological Plausibility

Finally I come to the controversial matter of what it means for a model to be "psychologically plausible", and since so much has been written and said on the matter, (e.g. Linell 1979) I will try to say as little as possible. Expressed with precaution, a model may be regarded as plausible if it adheres—on some feasible level—to the psychological evidence we have of human verbal behaviour.

Obviously psychological plausibility, as a goal for an NLP system, is qualitatively different from the other goals discussed above. Firstly it concerns only systems which explicitly try to model human verbal behaviour. A great many of them—openly or not—do not aspire to do so, be it for practical or theoretical reasons ("competence models" etc.). Then it seems that such a goal would take us back to the vagueness that often characterises approaches to theoretical linguistics and that we indirectly tried to avoid by setting up all "goals" and "criteria" up to now.

This is not quite so. When I said at the beginning that I was sceptical to such maximum goals as explanatory adequacy and psychological reality in the field of computational linguistics, I tried to stress—"to begin with". The point of laying down alternative goals was rather to give an idea of what properties formalisms or models should have in order to go beyond the limitations inherent in most current computational work with natural language. When the NLP frameworks have been developed to the degree that it becomes meaningful to ask questions pertaining matters such as plausibility and explanatory adequacy—then of course one should ask them.

TWOL seems to have been a breakthrough for computational morphology in this respect as well. One can still be critical towards it—for example concerning its stress on efficiency while somewhat neglecting perspicuity, but on the whole—as the discussion of properties 3.1–3.5 indirectly demonstrated—it provides reliable ground to build on.

Probably it has also been the first computational model of morphology to make an interesting hypothesis about human processing of morphological information. The evidence that Koskenniemi (1983) discusses is based on performance errors. It concerns the fact that speakers—mainly children and aphasics—tend to make mistakes in that they produce word-forms of more productive inflectional patterns instead of less productive ones. But errors that correspond to "automatic alternations", i.e. phonologically motivated ones, are extremely rare. So if one describes only this type of alternations with two-level rules, the following "error model" can be stated: "The performance errors in producing word forms are mostly faulty choices in the construction of the lexical level—there are hardly any errors in the application of the two-level rules" (Koskenniemi 1983:132). The simplicity and directness of the two-level rules in contrast with the intermediate stages and orderings of generative phonology rules make it possible to say that the TWOL framework actually **predicts** this error model. In this sense TWOL is more plausible than e.g. generative phonology.

4 Conclusions

In the preceding I have presented in a somewhat speculative way a number of goals for natural language processing systems. To make this less speculative I have concentrated on computational morphology and most of all on one single representative—the two-level model. The examples of Bulgarian morphology have been provided in the belief that "metatheoretical" studies in linguistics can not be carried on successfully without "anchorings" in particular languages. The choice of Bulgarian has been dictated by the fact that only there can I claim some originality.

The goals which I have brought up are of course not absolute, but I have held that they provide a good starting point for setting "standards" in computational linguistics and more specifically—in computational morphology. For each of these goals I have discussed properties of the frameworks that the systems are based on, which are necessary or simply desirable for the sake of these goals. The following table shows these once more,

GOALS of NLP SYSTEMS	PROPERTIES of NLP FRAMEWORKS
Sufficient coverage	Weak completeness
Efficiency	Restrictiveness
Augmentability	Perspicuity
	Declarativeness
	Modularity
Flexibility	Hard- and software independence
	Linguistic flexibility
Economy	Bidirectionality
Psychological plausibility	?

It is the goals that are the more stable part. Different systems can lay stress on some as opposed to others but still system-goals are less controversial and less contradictory to one another. The properties, however, can interact in complex ways and sometimes work against some other goal than the one which has called for them in the first place. So there is no exact relation between goals and properties, but rather a dynamic process giving priority to some instead of others in particular cases.

Despite that, I believe that these properties can be regarded as guidelines for work in the field of computational morphology and probably in computational linguistics as a whole, which is even more important in the absense of clear criteria for what should be considered as psychological plausibility and explanatory adequacy in the field of theoretical linguistics and the neigbouring disciplines. The question mark in the table above should signify that. Whatever its answer, I think that one could say that it is not "linguistic felicity". What is meant by this is as controversial as anything in linguistics nowadays. Unless, it is equivalent to what I have called (in 3.4.) "linguistic flexibility", i.e. the possibility of expressing language descriptions according to different types of theoretical models. However, I see this as a means for achieving the aim, not as the aim itself.

The whole point of working independently of theory is only to return to theory, but with less sectarianism and greater insight. Only this way will computational linguistics contribute to the understanding of what a psychologically plausible model should be, which is a prerequisite for approaching a model/theory of natural language that is "psychologically real". In doing so, computational linguistics is bound to come closer and probably merge with the broader paradigm of cognitive science.

References

Alam, Yukio Sasaki. 1983. A Two-level Morphological Analysis of Japanese. *Texas Linguistic Forum*, 22:229–252.

Bear, John. 1988. Morphology with Two-Level Rules and Negative Rule Features. *Proceedings of the 12th International Conference on Computational Linguistics*:28–31, COLING, Budapest.

Calder, Jonathan. 1989. Paradigmatic Morphology. *Proceedings of the Forth Conference of the European Chapter of the Association for Computational Linguistics*:58–65, ACL, Manchester.

Dalrymple, Mary, Lauri Karttunen and Sami Shaio. 1987. DKIMMO: A Morphological Analyzer using Two-Level Rules. In R. Kaplan and L. Karttunen: Computational Morphology, Course Script LI283, 1987 Linguistic Institute, Stanford University, June 29–August 7, 1987.

Gajek, Oliver, Hanno T. Beck, Diane Elder and Greg Whittemore. 1983. KIMMO Lisp Implementation. *Texas Linguistic Forum*, 22:187–202.

Kaplan, Ronald. 1987. Three seductions of computational psycholinguistics. In P. Whitelock et al. (Eds.), *Linguistic Theory and Computer Applications*:149–188, Academic Press, New York.

Karttunen, Lauri, Kimmo Koskenniemi and Ronald Kaplan. 1987. TWOL: A Compiler for Two-Level Phonological Rules. In R. Kaplan and L. Karttunen: Computational Morphology, Course Script LI283, 1987 Linguistic Institute, Stanford University, June 29–August 7, 1987.

Karttunen, Lauri and Kent Wittenburg. 1983. A Two-Level Morphological Analysis of English. *Texas Linguistic Forum*, 22:217–228.

Kataja, Laura and Kimmo Koskenniemi. 1988. Finite-state description of Semitic morphology: A case study of ancient Accadian. *Proceedings of the 12th International Conference on Computational Linguistics*:313–315, COLING, Budapest.

Khan, Robert. 1983. A Two-level Morphological Analysis of Rumanian. *Texas Linguistic Forum* 22:253–270.

Koskenniemi, Kimmo. 1983. *Two-level morphology: A general computational model for word-form recognition and production.* Publication 11, Department of General Linguistics, University of Helsinki, Helsinki.

Lindstedt, Jouko. 1984. A Two-level Description of Old Church Slavonic Morphology. *Scando-Slavica*, 30:165–189.

Linell, Per. 1979. *Psychological reality in phonology: A theoretical study.* Cambridge University Press, Cambridge.

Matthews, P. H. 1974. *Morphology.* Cambridge University Press, Cambridge.

Nyman, Martti. 1988. Abstrakta fonem och modeller som førstår bara "modellska". The Linguistic Association of Finland, Helsinki.

Sågvall-Hein, Anna, Mats Dahllöf and Sofia Hörmander. 1989. Studies of Grammars, Formalisms, and Parsing. *Rapporter från Språkdata*, 25. Department of Computational Linguistics, University of Gothenburg.

Shieber, Stuart. 1987. Separating Linguistic Analyses from Linguistic Theories. In U. Reyle and C. Rohrer (Eds.)*Natural Language Processing and Linguistic Theories*, D.Reidel, Holland.

Sproat, Richard and Barbara Brunson. 1987. Constituent-Based Morphological Parsing: A New Approach to the Problem of Word-Recognition. *Proceedings of the 25th Annual Meeting of the Association for Computational Linguistics*65–72, ACL, Stanford, CA.

Zlatev, Jordan. 1988. Bulgarian Nominal Inflection in the Two-Level Framework. Manuscript, Department of Linguistics, University of Stockholm.

Zlatev, Jordan, Gunnar Eriksson and Gunnel Källgren. 1989. *Linguistic Theories, Formalisms and Natural Language Processing: an Overview.* Institute of Linguistics, University of Stockholm.

Department of Linguistics
University of Stockholm
S-106 91 Stockholm
`JORDAN@COM.QZ.SE`